

Solving problems in parameter redundancy using computer algebra

E. A. CATCHPOLE¹, B. J. T. MORGAN² & A. VIALLEFONT^{3*},

¹Australian Defence Force Academy, Australia, ²University of Kent at Canterbury, UK and ³Laboratoire Sabres, Vannes, France

ABSTRACT *A model, involving a particular set of parameters, is said to be parameter redundant when the likelihood can be expressed in terms of a smaller set of parameters. In many important cases, the parameter redundancy of a model can be checked by evaluating the symbolic rank of a derivative matrix. We describe the main results, and show how to construct this matrix using the symbolic algebra package Maple. We apply the theory to examples from the mark-recapture field. General code is given which can be applied to other models.*

1 Introduction

1.1 Background

Let us suppose that a probability model has been proposed for a set of data, and that we intend to fit the model to the data using maximum likelihood. It is often the case that the likelihood surface is maximized on a completely flat ridge or plane, due to a redundancy in the parameter set. As we shall see from the examples later, it can be difficult to gauge whether or not all the parameters can, in principle, be estimated from the data. Areas in which this occurs include compartment modelling (Seber & Wild, 1989, chapter 8), Kalman Filter methodology (Harvey, 1989, p. 205), ion-channel modelling (Chen *et al.*, 1997), directed networks (Geiger *et al.*, 1996; Whitley, 1999), econometrics (Rothenberg, 1971), latent structure models (Goodman, 1974), and in models for the analysis of recovery/recapture data resulting from observations on marked animals (Freeman & Morgan, 1992; Lebreton *et al.*, 1992). If data are missing it may no longer be possible to

Correspondence: E. A. Catchpole, School of Mathematics and Statistics, University of New South Wales at ADFA, Canberra ACT 2600, Australia. E-mail: e.catchpole@adfa.edu.au

**Present address:* Institut Universitaire de Technologie, Lyon, France.

estimate all of the parameters in a model that is not parameter redundant (Catchpole & Morgan, 2001).

Recent research has shown how, for a wide class of models, established methods of computer algebra may be used to detect which models are parameter redundant (Catchpole & Morgan, 1997), and to determine which parameter combinations are estimable, i.e. have unique maximum-likelihood estimates (Catchpole *et al.*, 1998). After summarizing existing results for detecting parameter redundancy, we show how the basic problems may be solved in a simple and straightforward manner using the symbolic computation package, Maple. In order to do this, we focus on a single field, namely models for mark-recapture data. However, the procedures given apply equally well to other fields, using straightforward modifications to the Maple code provided here. Evidently, alternative symbolic algebra computer packages may be used.

Consider a data vector $\mathbf{y} = (y_1, \dots, y_n)$ from an exponential family distribution. The models we consider specify the distribution and provide an expression for the mean vector $\boldsymbol{\mu} = E[\mathbf{y}]$ in terms of a parameter vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_q)$, say. A model is parameter redundant if $\boldsymbol{\mu}$ can be expressed in terms of a parameter vector $\boldsymbol{\beta} = (\beta_1, \dots, \beta_r)$, with $r < q$. Otherwise it is said to be full rank. The test for parameter redundancy of Catchpole & Morgan (1997) requires the formation of the derivative matrix,

$$\mathbf{A} = \left\{ \frac{\partial \mu_i}{\partial \theta_j} \right\}, \quad 1 \leq j \leq q, \quad 1 \leq i \leq n \tag{1}$$

Catchpole & Morgan (1997) show that the model is parameter redundant if and only if \mathbf{A} is symbolically row-rank deficient, i.e. if and only if there exists a non-trivial vector function $\boldsymbol{\alpha}(\boldsymbol{\theta})$, such that

$$\boldsymbol{\alpha}(\boldsymbol{\theta})^T \mathbf{A} = 0 \quad \text{for all } \boldsymbol{\theta}. \tag{2}$$

It is equivalent (and often easier) to replace $\boldsymbol{\mu}$ by a suitable monotonic function, and $\log(\boldsymbol{\mu})$ is used in the examples below.

Models for mark-recapture data typically involve several independent multinomial distributions, and are examined for parameter redundancy in the same way as above.

If models are selected by means of information criteria (see, for example, Burnham & Anderson, 1998, and Lebreton *et al.*, 1992) it is important to know how many separate parameters have been estimated in the model-fitting. Current procedures for assessing parameter-redundancy involve making numerical approximations to the Hessian matrix of the log-likelihood (see, for example, Viallefont *et al.*, 1998). The test of equation (2) is equivalent to testing the rank of the expected Hessian matrix (Catchpole & Morgan, 1997). In the numerical approach, an assessment is made that combines the inherent structure of the model with particular aspects of the data set that is analysed. The work described in this paper provides a different approach, since we show how first to determine the parameter-redundancy of a model, and then how to gauge whether a data set that possesses missing values does or does not increase the parameter-redundancy.

The emphasis in this paper is on the use of a symbolic algebra computer package, for models and data sets of a particular size. However, in some cases, it is possible to draw conclusions by direct observation, rather than use symbolic algebra (see, for example, Catchpole *et al.*, 1996). Additionally, as we explain in Section 5, it is

desirable, and often possible, to establish the parameter redundancy of a particular model, irrespective of the size of the study. This is a useful and simplifying feature, which avoids a large amount of duplication of testing effort.

1.2 Models for mark-recapture data

In mark-recapture studies, animals are marked and released and it is recorded when (if at all) they are next seen/recaptured alive during the course of the study. Frequently, such studies work to an annual timescale that coincides with the life-cycle events of many wild animals.

Example 1: The Cormack-Jolly-Seber model

One of the basic models for such a study is the Cormack-Jolly-Seber model (Cormack, 1964; Jolly, 1965; Seber, 1965), with time-dependent annual survival probabilities $\{\phi_i\}$, and time-dependent annual recapture probabilities $\{p_i\}$. Suppose that there are $m = 2$ years of marking and release of the animals and $k = 3$ years of recapture, and let $\Omega_{i,j}$ be the probability that an animal marked in year i is next recaptured in year $j + 1$. The resulting matrix of recapture probabilities is

$$\Omega = \begin{bmatrix} \phi_1 p_1 & \phi_1 \phi_2 (1 - p_1) p_2 & \phi_1 \phi_2 \phi_3 (1 - p_1) (1 - p_2) p_3 \\ 0 & \phi_2 p_2 & \phi_2 \phi_3 (1 - p_2) p_3 \end{bmatrix}$$

The corresponding data matrix is known as the m -array (Burnham *et al.*, 1987, pp. 34 *et seq.*).

In order to check for parameter redundancy, it is not necessary to consider the multinomial cell probabilities corresponding to animals that are never captured during the study (see Catchpole & Morgan, 1997). This is a particularly simple model, and it is well known that, in the case $m = k$, the model is parameter redundant, since ϕ_k and p_k only appear in Ω as the product $\phi_k p_k$, but that all other parameters in the model are estimable. Even for this simple model, however, it is not clear whether all other parameters remain theoretically estimable in a case such as the above where $m < k$. We show how to answer this question using simple Maple code, and also illustrate more complex models, for which it is not otherwise easy to decide the parameter redundancy.

Mark-recapture models are often very much more complicated than that of Example 1, frequently involving a large number of parameters. One illustration is when animals may be distributed over different sites, and move between those sites.

2 The probability matrix

Once a model is specified, we need to construct the probability matrix Ω . It can save a lot of time, and eliminate the possibility of errors, if this is done using computer algebra. We shall now consider how to proceed; more powerful features of Maple will be used later in the paper. Example 1 has both the survival probabilities ϕ and the recapture probabilities p being purely time-dependent. This can be represented symbolically by the parameter matrices

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \\ 0 & \phi_2 & \phi_3 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} p_1 & p_2 & p_3 \\ 0 & p_2 & p_3 \end{bmatrix}$$

where $\Phi_{i,j}$ is the probability that an animal from cohort i , alive at occasion j , survives until occasion $j+1$, and $P_{i,j}$ is the probability that an animal from cohort i , alive at $j+1$, is recaptured then. The Maple code to set up these matrices is as follows.

```
> m := 2: k := 3:
> phi := vector(k): p := vector(k):
> Phi := matrix(m,k,0): P := matrix(m,k,0):
> for i from 1 to m do
  for j from i to k do
    Phi[i, j] := phi[j]: P[i, j] := p[j]:
  od;
od;
```

A simpler method of denoting this pure time dependence is to use the parameter index matrices (PIM)

$$\Phi_{\text{pi}} = P_{\text{pi}} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 3 \end{bmatrix}$$

This method is used in the MARK package (White & Burnham, 1999) for the analysis of mark-recapture and recovery data. After constructing the matrix P_{pi} as above, all that is then required is

```
> Phi := Index2Mat(Ppi, phi): P := Index2Mat(Ppi, p):
```

where the code for `Index2Mat` is given in the Appendix. A simpler method still does not use PIM, but instead implements special procedures for particular models. For purely time-dependent parameters, as above, we could use

```
> Phi := Time(phi): P := Time(p):
```

using the procedure `Time` given in the Appendix.

The probability matrix Ω is then constructed in two stages. First, the procedure `CumSurviv` (cumulative survival) transforms the matrix Φ into the matrix

$$\text{CumSurviv}(\Phi) = \begin{bmatrix} \phi_1 & \phi_1 \phi_2 & \phi_1 \phi_2 \phi_3 \\ 0 & \phi_2 & \phi_2 \phi_3 \end{bmatrix}$$

and `CumRecap` transforms \mathbf{P} into

$$\text{CumRecap}(P) = \begin{bmatrix} p_1 & (1-p_1)p_2 & (1-p_1)(1-p_2)p_3 \\ 0 & p_2 & (1-p_2)p_3 \end{bmatrix}$$

Secondly, these two matrices are multiplied together, elementwise using `pmult`, to form Ω . The Maple code for this operation is simply

```
Omega := pmult(CumSurviv(Phi), CumRecap(P));
```

The code for `CumSurviv`, `CumRecap` and `pmult` is also given in the Appendix.

It should be clear now how other standard models, incorporating age-dependence, for example, may be similarly programmed, as well as complex models tailored to particular data. The beauty of this use of Maple for constructing Ω is that it reduces the chance of human error at this stage.

3 The derivative matrix

3.1 The rank of \mathbf{A}

In order to form the derivative matrix \mathbf{A} , we collect the non-zero elements of the probability matrix Ω into a single vector ω , take logarithms of the elements ω , and then calculate the symbolic derivatives with respect to the parameters. The rank of the resulting matrix then gives the number of parameters that can be estimated by maximum likelihood. This is done by means of the following commands, with the results shown for the model of Example 1.

```
> q := vectdim(phi) + vectdim(p);
```

```
q:=6
```

```
> A := Dmat(Omega,phi,p);
```

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\phi_1} & \frac{1}{\phi_1} & \frac{1}{\phi_1} & 0 & 0 \\ 0 & \frac{1}{\phi_2} & \frac{1}{\phi_2} & \frac{1}{\phi_2} & \frac{1}{\phi_2} \\ 0 & 0 & \frac{1}{\phi_3} & 0 & \frac{1}{\phi_3} \\ \frac{1}{p_1} & -\frac{1}{1-p_1} & -\frac{1}{1-p_1} & 0 & 0 \\ 0 & \frac{1}{p_2} & -\frac{1}{1-p_2} & \frac{1}{p_2} & -\frac{1}{1-p_2} \\ 0 & 0 & \frac{1}{p_3} & 0 & \frac{1}{p_3} \end{bmatrix}$$

```
> with(linalg): rank(A);
```

```
4
```

Note that in this small example we have illustrated the derivative matrix \mathbf{A} . This will not normally be shown. The code for `Dmat` is given in the Appendix.

Since $\text{rank}(\mathbf{A}) = 4$ is less than the number of parameters $q = 6$, the model is parameter redundant, from Catchpole & Morgan (1997). Since the deficiency, defined as $n - \text{rank}(\mathbf{A})$, is 2 rather than 1, it is not just the elements of the combination $\phi_3 p_3$ that are non-estimable: the rank shows that there are only four independent theoretically estimable combinations of the six parameters $\phi_1, \phi_2, \phi_3, p_1, p_2, p_3$, from Catchpole *et al.* (1998).

3.2 Theoretically estimable parameters

When a model is parameter redundant, it is important to know which, if any, of the parameters are theoretically estimable. To discover this, we need to consider the general solution vector α to (2), as explained in Catchpole *et al.* (1998). This is accomplished by the instructions below.

```
> zero := vector(coldim(A), 0):  
> alpha := linsolve(transpose(A), zero, 'r', t);
```

$$\alpha = \left[0, \frac{\phi_2(-t_1 p_3 + t_1 p_3 p_2 - \phi_3 t_2 + t_2 \phi_3 p_2)}{\phi_3 p_3}, t_1, 0, \right. \\ \left. - \frac{p_2(-t_1 p_3 + t_1 p_3 p_2 - \phi_3 t_2 + t_2 \phi_3 p_2)}{\phi_3 p_3}, t_2 \right]$$

This form of the `linsolve` command assigns $\text{rank}(\mathbf{A})$ to the variable r and uses t for any unknown constants. The vector α has two arbitrary constants, t_1 and t_2 , since the model has deficiency 2. However α has zero entries in positions corresponding to the rows in \mathbf{A} resulting from taking derivatives with respect to the parameters ϕ_1 and p_1 . By Catchpole *et al.* (1998), therefore, ϕ_1 and p_1 are theoretically estimable, but none of the other parameters are.

There must exist two more independent theoretically estimable combinations of the parameters, since $\text{rank}(\mathbf{A}) = 4$. These can be found, as explained in Catchpole *et al.* (1998), by solving the set of linear first-order partial differential equations

$$\sum_{s=1}^q \alpha_{s,j} \frac{\partial f}{\partial \theta_s} = 0, \quad j = 1, \dots, d \tag{3}$$

where in this case there are $q = 6$ parameters and the deficiency is $d = 2$. Here we are denoting by α_1 and α_2 the independent solutions of (2) formed by taking $t_2 = 0$ and $t_1 = 0$ respectively, and letting $\alpha_{s,j}$ be the s th component of α_j .

The pair of equations (3) can be solved using Maple, although we do not show the code here (see Gimenez, 2001). In the example above, we can in fact see by inspection that the matrix Ω can be written in terms of ϕ_1 , p_1 , $\phi_2 p_2$ and $\phi_2 \phi_3 (1 - p_2) p_3$. Since these are clearly independent parameter combinations, in the sense that no one can be obtained from the others, they must be the four independent theoretically estimable parameter combinations. For more complex examples, identification of theoretically estimable parameter combinations by inspection is likely to be more difficult than here.

3.3 Missing data

Although a model may be full rank, missing data may render certain parameters inestimable in practice in any particular application (Catchpole & Morgan, 2001). Suppose for example, in Example 1, no animals were recaptured from the first cohort of marked animals in year 3 of the study. This gap in the m -array results in the element $\Omega_{1,3}$ not appearing in the likelihood, and the derivative matrix must be amended so that this element is omitted. No other changes are required in the code, except that it now becomes essential to add an extra column to Ω to incorporate probabilities for animals not recaptured at all during the study, so that each row becomes a full multinomial distribution (see Catchpole & Morgan, 2001). The Maple code for this illustration is shown below.

```
> X := matrix(m,1):  
> for i to m do X[i,1] := 1 - sum(Omega[i,j], j = 1..k) od;
```

```

> Omega_X := augment(Omega,X):
> Omega_X[1,3] := 0: print(Omega_X);


$$\begin{bmatrix} \phi_1 p_1 \phi_1 \phi_2 (1-p_1) p_2 & 0 & 1 - \phi_1 p_1 - \phi_1 \phi_2 (1-p_1) p_2 - \phi_1 \phi_2 \phi_3 (1-p_1) (1-p_2) p_3 \\ 0 & \phi_2 p_2 & \phi_2 \phi_3 (1-p_2) p_3 & 1 - \phi_2 p_2 - \phi_2 \phi_3 (1-p_2) p_3 \end{bmatrix}$$


> A_X := Dmat(Omega_X, phi, p):
> rank(A_X);

```

4

The rank of \mathbf{A} is unchanged, so that in this case the missing data do not affect the parameter redundancy.

4 Example 2: two groups of animals

Consider now a situation in which there are two groups of animals (e.g. two sexes), with time-dependent survival and recapture probabilities, as in Example 1, but with the survival and recapture probabilities of group 2 being constant multiples of those for group 1. Thus, the survival probabilities for groups 1 and 2 can be written as Φ_1 and Φ_2 , where Φ_1 is as in Example 1 and $\Phi_2 = a\Phi_1$, for some constant a ; and similarly for the recapture probabilities we have $P_2 = bP_1$, for some constant b . In the notation of Lebreton *et al.* (1992), the models for both survival and recapture are of the form ‘time + group’, since the group effect is additive on a logarithmic scale.

In examples such as this, it is convenient to have a separate probability matrix Ω for each group. To construct the overall derivative matrix, it is then sufficient to arrange these probability matrices side-by-side, using the Maple `augment` command. The code required in this illustration is:

```

> m:=2:k:=3: phi:=vector(k): p:=vector(k): a:='a': b:='b':
> Phi1:=Time(phi); # survival for group 1

```

$$\Phi_1 := \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \\ 0 & \phi_2 & \phi_3 \end{bmatrix}$$

```

> Phi2:=evalm(a*Phi1); # survival for group 2

```

$$\Phi_2 := \begin{bmatrix} a\phi_1 & a\phi_2 & a\phi_3 \\ 0 & a\phi_2 & a\phi_3 \end{bmatrix}$$

```

> P1:=Time(p): P2:=evalm(b*P1):
> Omega1:=pmult(CumSurviv(Phi1), CumRecap(P1)):
> Omega2:=pmult(CumSurviv(Phi2), CumRecap(P2)):
> Omega:=augment(Omega1,Omega2):
> q:=vectdim(phi)+vectdim(p)+2;

```

$$q:=8$$

```

> A:=Dmat(Omega,phi,p,a,b): rank(A);

```

7

```
> zero := vector(coldim(A), 0):  
> alpha := linsolve(transpose(A), zero, 'r', t);
```

$$\alpha := \left[0, 0, -\frac{t_1 \phi_3}{p_3}, 0, 0, t_1, 0, 0 \right]$$

The model is therefore parameter redundant, since $\text{rank}(\mathbf{A}) = 7$ is less than the number of parameters, $q = 8$. Unlike Example 1, the deficiency is now only 1. Furthermore, since α has zeros in every position except those corresponding to ϕ_3 and p_3 , all parameters except these two are estimable. It can be seen by inspection that the other estimable parameter combination is the product $\phi_3 p_3$. An interesting aspect of this model is that if the group effect is additive *on any other scale*, such as the logistic for example, then the model is not parameter redundant (Viallefont, 1995, ch. 3). We hypothesize that in such a case the model will be ‘near-redundant’, using the terminology of Catchpole *et al.* (2001), and may result in certain parameters being estimated with low precision.

Near-redundant models are not parameter redundant. However, they may provide poor estimates of some model parameters, as a result of having small eigenvalues of the information matrix. Catchpole *et al.* (2001) suggest that the numerical procedure of Viallefont *et al.* (1998) is then needed.

Choquet (2001) and Gimenez (2001) provide applications to multi-state mark-recapture models.

5 Discussion

In the examples we have given, we have used very small studies, with $m = 2$ years of marking and $k = 3$ years of recaptures. We have done this purely for illustration. There are, in principle, no problems in using Maple on much larger problems. Focusing on a derivative matrix, rather than an expected Hessian matrix, greatly enhances the speed of the Maple procedures. Extension theorems, mentioned below, quite often result in Maple only being needed for ‘small’ examples of models of a given structure. Furthermore we have also, for pedagogical reasons, kept the models considered to be fairly simple. However, the methods given can very easily be applied to test quite complicated models, including for example all those considered by Lebreton *et al.* (1992).

Any results obtained by the methods described above are for a fixed size of study only. It would clearly be beneficial to be able to draw general conclusions about all studies where the survival and recapture models are of a particular type (e.g. both purely time-dependent). In other words, we would like to extrapolate from the results for a particular m and k to general m and k . Catchpole & Morgan (1997) give such an extension theorem for the full rank case; that is, they show that, under suitable conditions, if a model is full rank for a small study then it will remain full rank for a larger study. The corresponding theory for the parameter-redundant case, showing when the deficiency is preserved, is given in Catchpole & Morgan (2001). An illustration is provided by Catchpole & Morgan (2001), in which it is supposed that in the Cormack-Jolly-Seber model there is an immediate effect of capture, resulting in a change in the probability of capture that extends to the following year only. It is shown that when $m = k \geq 4$, the model has deficiency 1. Additional examples are provided by Kgosi (2000). Currently, little is known for certain when simple models are extended to account for age and/or capture effects.

The procedures of this paper can be used with confidence to provide definite statements regarding parameter-redundancy, and for particular data sets to investigate whether missing data values may modify the parameter-redundancy.

The use of Maple to calculate the symbolic rank of the derivative matrix relies on this matrix being a rational function of the parameters. Although this covers many examples of practical interest, it does not cover all cases. One important exception would be where parameters appear in a non-linear way, for example in describing the dependence of survival on a covariate. The possibility of using computer algebra in such situations is under current investigation. For the moment, it is necessary in such cases to revert to numerical methods, as in Viallefont *et al.* (1998).

REFERENCES

- BURNHAM, K. P. & ANDERSON, D. R. (1998) *Model Selection and Inference. A Practical Information-theoretic Approach* (New York, Springer).
- BURNHAM, K. P., ANDERSON, D. R., WHITE, G. C., BROWNIE, C. & POLLOCK, K. H. (1987) Design and analysis methods for fish survival experiments based on release-recapture. American Fisheries Society, Monograph 5, Bethesda, Maryland.
- CATCHPOLE, E. A., FREEMAN, S. N. & MORGAN, B. J. T. (1996) Steps to parameter redundancy in age-dependent recovery models, *Journal of the Royal Statistical Society B*, 58, pp. 763-774.
- CATCHPOLE, E. A., KGOSI, P. M. & MORGAN, B. J. T. (2001) On the near-singularity of models for animal recovery data, *Biometrics*, 57, pp. 720-726.
- CATCHPOLE, E. A. & MORGAN, B. J. T. (1997) Detecting parameter redundancy, *Biometrika*, 84, pp. 187-196.
- CATCHPOLE, E. A. & MORGAN, B. J. T. (2001) Deficiency of parameter-redundant models, *Biometrika*, 88, pp. 593-598.
- CATCHPOLE, E. A., MORGAN, B. J. T. & FREEMAN, S. N. (1998) Estimation in parameter redundant models, *Biometrika*, 85, pp. 462-468.
- CHEN, D., LEAR, J. & EISENBERG, B. (1997) Permeation through an open channel: Poisson-Nernst-Planck theory of a synthetic ionic channel, *Biophysical Journal*, 72, pp. 97-116.
- CHOQUET, R. (2001) Computing easily the rank of multi-state capture-recapture models.
- CORMACK, R. M. (1964) Estimates of survival from the sighting of marked animals, *Biometrika*, 51, pp. 429-438.
- FREEMAN, S. N. & MORGAN, B. J. T. (1992) A modelling strategy for recovery data from birds ringed as nestlings, *Biometrics*, 48, pp. 217-236.
- GEIGER, D., HECKERMAN, D. & MEEK, C. (1996) Asymptotic model selection for directed networks with hidden variables. Technical Report MSR-TR-96-07, Microsoft Research, Redmond WA 98052, USA.
- GIMENEZ, O. (2001) Parameter redundancy for multistate capture-recapture models.
- GOODMAN, L. A. (1974) Exploratory latent structure analysis using both identifiable and unidentifiable models, *Biometrika*, 61, pp. 215-231.
- HARVEY, A. C. (1989) *Forecasting, Structural Time Series and the Kalman Filter* (Cambridge University Press).
- JOLLY, G. M. (1965) Explicit estimates from capture-recapture data with both death and immigration-stochastic models, *Biometrika*, 52, pp. 225-247.
- KGOSI, P. M. (2000) Models for avian demography. PhD thesis, University of Kent at Canterbury. Unpublished.
- LEBRETON, J.-D., BURNHAM, K. P., CLOBERT, J. & ANDERSON, D. R. (1992) Modeling survival and testing biological hypotheses using marked animals: a unified approach with case studies, *Ecological Monographs*, 62, pp. 67-118.
- ROTHENBERG, T. J. (1971) Identification in parametric models, *Econometrica*, 39, pp. 577-591.
- SEBER, G. A. F. (1965) A note on the multiple recapture census, *Biometrika*, 52, pp. 249-259.
- SEBER, G. A. F. & WILD, C. J. (1989) *Nonlinear Regression* (New York, Wiley).
- VIALLEFONT, A. (1995) Robustesse et flexibilité des analyses démographiques par capture-recapture: de l'estimation de la survie à la détection de compromis évolutifs. PhD thesis, University of Montpellier II. Unpublished.
- VIALLEFONT, A., LEBRETON, J.-D., REBOULET, A.-M. & GORY, G. (1998) Parametric identifiability

and model selection in capture-recapture models: a numerical approach, *Biometrical Journal*, 40, pp. 313-325.

WHILEY, M. (1999) Aspects of the interface between statistics and neural networks. PhD thesis. University of Glasgow. Unpublished.

WHITE, G. C. & BURNHAM, K. P. (1999) Program MARK: survival estimation from populations of marked animals, *Bird Study*, 46 (suppl.), pp. 120-139.

Appendix

Maple procedures

The following code is available from www.ma.adfa.edu.au/~eac/Redundancy/Maple.

A1 Index2Mat

```
Index2Mat := proc(Indx,theta)
# Calculates a parameter matrix from the parameter index matrix Indx
# using the parameter vector theta
local i,j,M,m,k;
m := rowdim(Indx): k := coldim(Indx): M := matrix(m,k,0):
  for i to m do
    for j from i to k do
      if Indx[i,j] = 0 then M[i,j] := 0:
      else M[i,j] := theta[Indx[i,j]]:
      fi
    od;
  od;
evalm(M);
end:
```

A2 Time

```
Time := proc(p)
# Calculates a time-dependent parameter matrix using parameter vector p
local i,j,P;
P := matrix(m,k,0):
  for i to m do
    for j from i to k do
      P[i,j] := p[j]:
    od;
  od;
evalm(P);
end:
```

A3 CumSurviv

```

CumSurviv := proc(P)
# Converts a parameter matrix to a cumulative survival matrix
local i,j,CP:
CP := evalm(P):
for i to rowdim(P) do
  for j from i+1 to coldim(P) do
    CP[i,j] := product(P[i,t],t=i..j);
  od;
od;
evalm(CP);
end:

```

A4 CumRecap

```

CumRecap := proc(P)
# Converts a parameter matrix to a cumulative recapture matrix
local i,j,CP:
CP := evalm(P):
for i to rowdim(P) do
  for j from i+1 to coldim(P) do
    CP[i,j] := product(1-P[i,t],t=i..j-1)*P[i,j]:
  od;
od;
evalm(CP);
end:

```

A5 pmult

```

pmult := proc(mat1,mat2)
# Pointwise multiplication of any number of matrices
local argno,i,j,m,n,A1,A2;
m := rowdim(args[1]): n := coldim(args[1]); A1 := evalm(args[1]):
for argno from 2 to nargs do
  if rowdim(args[argno]) = m and coldim(args[argno]) = n then
    for i to m do
      for j to n do
        A1[i,j] := A1[i,j]*args[argno][i,j]
      od
    od;
  else RETURN('All matrices must have the same dimensions')
fi;
od;
evalm(A1);
end:

```

A6 Dmat

```

Dmat := proc(P, th1, th2)
# Calculates the derivative matrix of the probability matrix P
# with respect to the parameters th1, th2, ...,
# which may be a mixture of vectors and scalars
local i,j,r,D,omega,m,k,nrow,ncol,nv,th;
m := rowdim(P): k := coldim(P):
# first string out all the parameters into a 1 x nrow matrix
if type(args[2],scalar) then nv := 1:
else nv := vectdim(args[2]);
fi;
th := matrix(1,nv,args[2]):
for i from 3 to nargs do
  if type(args[i],scalar) then
    nv := 1:
    th := augment(th,matrix(1,1,1));
    th[1,coldim(th)] := args[i];
  else
    nv := vectdim(args[i]):
    th := augment(th, matrix(1,nv,args[i]));
  fi;
od;
nrow := coldim(th):
# Now string out the nonzero probs into a vector omega of length ncol
i := 'i': ncol := 0:
for i to m do
  for j to k do
    if P[i,j] <> 0 then ncol := ncol+1:
    fi
  od;
od;
omega := vector(ncol);
i := 'i': j := 'j': r := 0:
for i to m do
  for j to k do
    if P[i,j] <> 0 then
      r := r+1:
      omega[r] := P[i,j]:
    fi
  od;
od;
# Now calculate the nrow x ncol derivative matrix D
D := array(1..nrow,1..ncol):
i := 'i': j := 'j':
for i to nrow do
  for j to ncol do
    D[i,j] := diff(log(omega[j]),th[1,i]):
  od;
od;
evalm(D);
end:

```